Middle-East Journal of Scientific Research 23 (Sensing, Signal Processing and Security): 265-268, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.ssps.106

VLSI Implementation of Low Power FIR Filter Design Using APC-OMS Algorithm

[1]S. Rajkumar, [2]P. Balaji and [3]Mrs. M.A. Lekshmidevi

[1]ECE Department, Arulmigu Meenakshi Amman College of Engineering, Vadamavandal, India
[2]EIE Department, St. Peters University, Avadi, Chennai, India
[3]EEE Department, St.Peters College of Engineering and technology, Avadi,Chennai, India

Abstract: Memory based structures are used in many kind of digital signal processing (DSP) application. Memory-based structures are better performance In area minimization compare with multiply-accumulate structures and have many other advantages like reduced latency since the memory-access-time is much shorter than the usual multiplication-time compared to the conventional multipliers. The multiplier uses LUT's as memory for their computations. The anti-symmetric product coding (APC) and odd- multiple-storage (OMS) techniques were used for look-up-table (LUT) in adaptive FIR filter. Memory-based structure such as APC and OMS techniques are used for efficient Multiplication. Hence, the combination of these two techniques provides reduction in LUT size to one fourth in adaptive FIR filter when compared with the conventional Look up Table (LUT) of adaptive FIR filter.

Key words: Finite impulse response filter · Look- Up Table(LUT) · Anti-symmetric Product Coding (APC) · Odd Multiple Storage(OMS) · Digital Signal Processing

INTRODUCTION

Finite impulse response (FIR) digital filters are common components in many digital signal processing (DSP) systems [1-10]. Throughout the years, with the increasingly development in very large scale integration (VLSI) technology, the real time realization of FIR filter with less hardware requirement and less latency has become more and more important. Because the complexity of implementation grows with the length of filter, several algorithms have been made to develop effective architectures for realization of FIR filters in application specific integrated circuits (ASIC) and field programmable gate arrays (FPGA) platforms and one of them is distributed arithmetic (DA). The main portion of DA-based computation is lookup table (LUT) that stores the pre-computed values and can be read out easily, which makes DA-based computation well suited for FPGA realization, because the LUT is the basic components of FPGA.

Moreover, this technology represents a number of attractive features such as simplicity, regularity and modularity of architecture. Also, the DA technique can be designed to meet various speed requirements, for example, it can be designed for high-speed implementation where all bits of one word are processed per clock, it can also be designed for medium- speed implementation where several bits of one word (not all bits) are processed per clock. In recent years, DA-based FIR filter has gained substantial popularity as a primary DSP operation and are rapidly replacing classic analog filters.

Algorithm: DA is an important FPGA technology. We briefly outline here the conventional DA approach for inner product computation [11, 13]. Here is the detail of DA. To understand the DA design paradigm [11], consider the ''sum of products'' inner product shown below

$$y = \sum_{n=0}^{N-1} A_n * x_n \tag{1}$$

Assume further that An is known constants and xn is a variable [11]. An unsigned DA system assumes that the xn is represented by

$$x_n = \sum_{b=0}^{B-1} 2^b x_{n,b} \quad \text{with} \quad x_{n,b} \in [0,1] \tag{2}$$

Where xn,b denotes the bth bit of xn, the nth sample of x [11]. The inner product can, therefore, be represented as

$$y = \sum_{b=0}^{B-1} 2^b \sum_{n=0}^{N-1} A_n x_{n,b} \tag{3}$$

Corresponding Author: Dr. S. Rajkumar, ECE Department, Arulmigu Meenakshi Amman College of Engineering, Vadamavandal

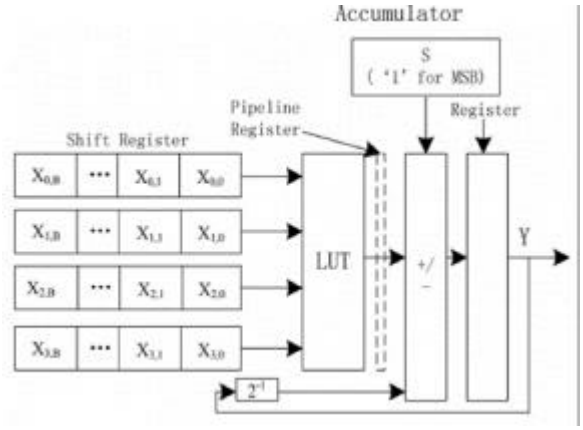

Fig. 1: FIR filters implementation with single LUT.

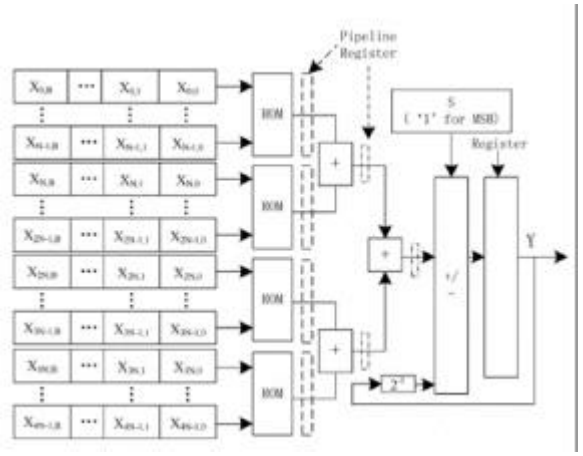

Fig. 2: FIR filters implementation with decomposed LUT's.

Implementation of the function Anxn,b requires special attention [11]. The preferred implementation method is to realize the mapping Anxn,b using one LUT [11]. That is, a 2N-word LUT is preprogrammed to accept an N-bit input vector xb and output Anxn,b [11]. The individual mappings Anxn,b are weighted by the appropriate power-of-two factor and accumulated.

After N look-up cycles, the inner product y is computed. Finally, the detail of LUT is shown in Table 1. Moreover, Fig. 1 [11] shows an original LUT-based DA implementation of a 4-order FIR filter.

Modified DA Solutions: Following the algorithm suggested above, we derive here the fully pipelined architecture with table partitioning for FIR filters in high-speed and medium-speed. As described above, the structures in Ref. [8] have many latency as well as the number of adders. Therefore, here we make a modification on them to reduce the latency and the number of adders. First of all, the boundary adders of the two-dimensional (2-D) structure of Ref. [8] can be replaced by the pipelined

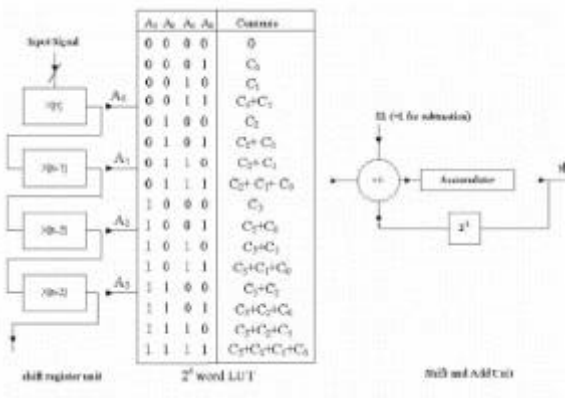

Fig. 3: Coeffcients update in Decomposed LUT.

shift-adder tree. In the 2-D structure of Ref. [18], there are B number of shift adders in the boundary line (B is the word length of the input sample) and therefore the latency is B cycle periods.

If the boundary shift adders are replaced by the pipelined shift-adder tree, there will be B-1 number of adders and the latency is log2 B cycle periods. Thus, the latency of the 2-D structure in Ref. [8] is reduced significantly, especially when B is a large number. And the modified architecture is shown in Fig. 1. Moreover, we can also change the systolic array adders of the architecture into the pipelined adder trees, as shown in Fig. 2. Therefore, the whole latency of the architecture is reduced significantly, as well as the number of adders. Since the architecture in Fig. 4 processes all bits of one input word per clock, the architecture can be used for high speed implementation.

LUT less FIR Implementation: The architecture proposed in this paper is shown in Fig. 4. In this architecture, the tri-state buffer and a carry lookahead adder are the basic digital logic units used to construct the on-line LUT. From this architecture it is clear that filter coefficients will pass to the CLA only if their buffer enable signal value is 1.

A conventional lookup-table (LUT)-based multiplier is showninFig.1,where A is a fixed coefficient and X is an input word to be multiplied with A. Assuming X to be a positive binary number of word length L, there can be 2L possible values of X and accordingly, there can be 2L possible values of product C = A • X. Therefore, for memory-based multiplication, an LUT of 2L words, consisting of pre computed product values corresponding to all possible values of X,is conventionally used.

Advantages of the On-Line DALUT Architecture: The advantages of the proposed architecture over the basic DA shown in Fig. 1 are summarized as follows.

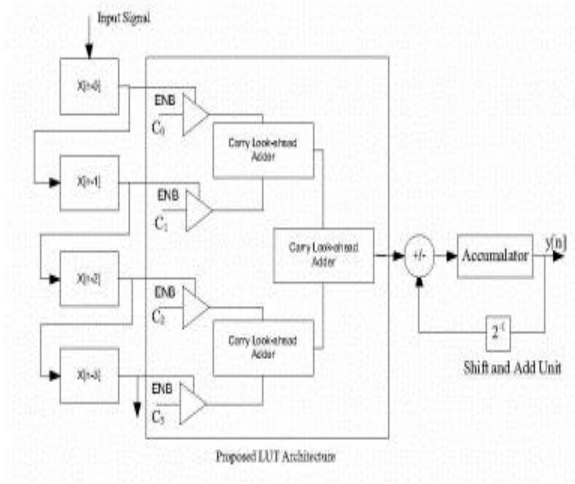

Fig. 4: LUT less FIR implementation.

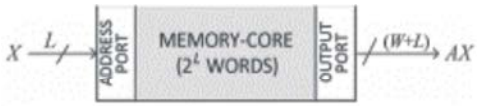

Fig. 5: Conventional LUT.

- Avoid the construction of large size ROM especially when the number of input data is high.
- Only the needed location contents are calculated whereas, in the basic DA technique the contents of locations that may not be used when processing the input signal are also computed.
- Avoid repetitive memory accessing which affects the system performance i.e. long processing time. the total number of memory accesses is given in (8).

$$Tm.a=N.K \quad (4)$$

Where T_{ma} the total number of memory accesses, N is the number of input variables and K is the word length of each filter input.

- In basic DA technique, even if the location content is zero it will be fetched and added to the partial sum, whereas in our on- line LUT no addition operation occurs when calculated contents is zero. This will guarantee that the execution time for obtaining the filter output will be as short as possible

Speed and Hardware Optimization: The use of tri-state buffer and CLA as basic logic units in the proposed architecture imposes a remarkable improvement in speed and area-time efficiency parameters. Fig. 6 shows buffer if 1 (bufifl) primitive with four outputs.

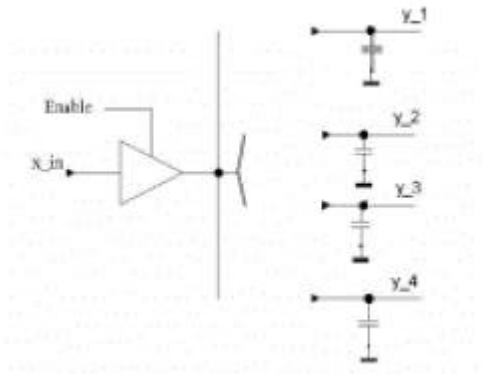

Fig. 6: Buffer if 1 instantiation.

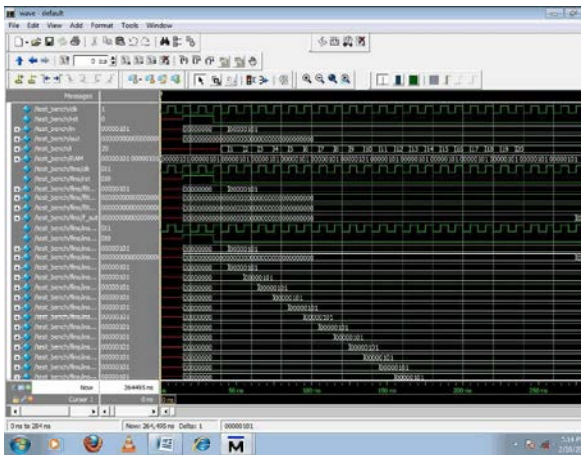

Fig. 7: Simulated output.

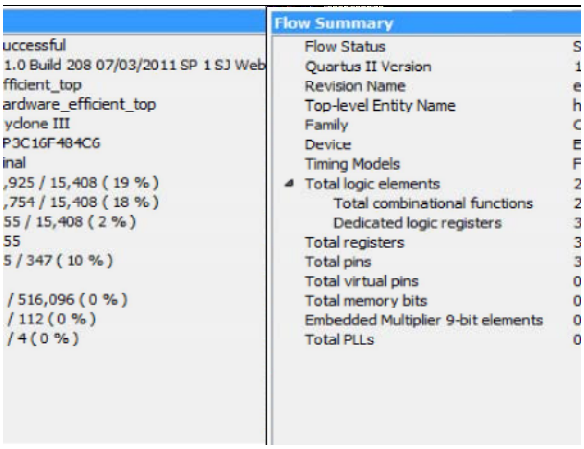

Fig. 8: Area utilization report.

Here when the enable is asserted, the outputs are determined by the input x in. Therefore, the enable signal which is any of the shift registers output bit, in Fig. 4, is a primary condition whether to transfer its associated filter coefficient to the CLA in the next stage or not. This is the main concept of the DA technique presented in [6].

According to [12], the number of transistors needed to construct 2 x 1 MUX is 6, whereas only 4 are needed for the tri-state buffer. Therefore, the use of tri-state buffer in the proposed architecture results in a reduction in the transistor count needed for the design compared to Fig. 3. Furthermore, the three- state gates can be used as primitive units in designing 2xl MUX [13]. Fig. 6 shows this architecture.

CONCLUSION

Analyzing the DA architecture and performance, this paper presents a new architecture for DALUT. The proposed architecture applies the main concept of the basic DA technique in implementing the MAC unit and at the same time has many advantages over its basic architecture. The prime logic units used in our proposed architecture is the carry lookahead adder and the tri-state buffer. The results obtained show that with the proposed architecture, the computation time and the area used is reduced. The proposed architecture can be easily used to implement high order FIR filters e.g. 20-tap with different coefficients wordlength without suffering from large LUT construction and with low hardware complexity needed for the design. The future work is to perform a VLSI implementation of pulse shaping FIR filter for ultra wideband communications.

REFERENCES

1. Antonion, A., 1993. Digital Filters: Analysis, Design and Applications, McGraw-Hill, New York.
2. Kung, H.T., 1982. Why systolic architecture? IEEE Computer 15(1): 37-45.
3. Yu, S. and E.E. Swartzlander, 2001. DCT implementation with distributed arithmetic, IEEE Transactions on Computers, 50(9): 985-991.
4. Hanho Lee and Gerald E. Sobelman, 2002. FPGA-based digit-serial CSD FIR filter for image signal format conversion, Microelectronics Journal, 33(5-6): 501-508.
5. Valeria Garofalo, 2008. Fixed-width multipliers for the implementation of efficient digital FIR filters, Microelectronics Journal, 39(12): 1491-1498.
6. Lei Zhang and Tadeusz Kwasniewski, 2009. FIR filter optimization using bit-edge equalization in high-speed backplane data transmission, Microelectronics Journal, 40(10): 1449-1457.
7. Eshtawie, M.A.M. and M. Othman, 2006. On-line DA-LUT architecture for high-speed high-order digital FIR filters, in: Proceedings of the IEEE International Conference on Communication Systems (ICCS), Singapore, November.
8. Choi, J.P., S.C. Shin and J.G. Chung, 2000. Efficient ROM size reduction for distributed arithmetic, in: Proceedings of the IEEE International Symposium Circuits Systems (ISCAS), May 2000, pp: 61-64.
9. Sanjay, Attri, B.S., Sohi and Y.C. Chopra, 2001. Efficient design of application specific DSP cores using FPGAs, in: International Conference on ASIC Proceedings, pp: 462-466.
10. Kim Kyung-Saeng and Kwyro Lee, 2003. Low-power and area efficient FIR filter implementation suitable for multiple tape, IEEE Transactions on VLSI Systems, 11(1) (February 2003).